Manual on tissue model, for simulation of spectral filtering

Josefine Holm Nielsen

September 6, 2015

1 Monte Carlo

The 3D Monte Carlo code we are using is based on a code written by Steven Jacques from the Oregon Health and Science University, which can be downloaded at http://omlc.ogi.edu/software/mc/mcxyz/index.html, the code was downloaded 07-01-2014.
In general this is a very flexible code which can model any type of structure. One thing to be aware of though is that the code cannot handle refractive index changes so it assumes the entire structure has a constant refractive index. This means it can't handle boundary reflections, and therefore any total internal reflection is ignored.
In tissue all the refractive indices are close to 1.5 so total internal reflection should be rare, since the critical angle is close to 90 (where 0 is the normal vector for the boundary).
One should also be aware that the Monte Carlo code uses some somewhat strange boundary conditions. If a photon escapes the matrix containing the tissue parameters it will be moved to the closes array position, which will always be on the boundary, and absorbed. This will sometimes lead to unusually high absorption just on the boundary compared to any internal position. The simple solution was made of simply setting the absorption on the boundary to zero, thereby assuming any photon escaping the system is thrown out.
The Monte Carlo code has been edited so that it could give a uniformly distributed light input over the entire surface. Contrary to what Steven says in the comments in the code it can only use a point source or a circle of uniform light.
The Monte Carlo code needs a very specific type of input file which is generated using the maketissue.m code. Within this MatLab file the user needs to:

- Build a matrix describing which types of tissue is where.
- Generate a matrix containing the optical properties of the tissue: scattering, μ_s [1/cm], absorption, μ_a [1/cm] and anisotropy, g.
- Describe how the input light is distributed.
- Determine how long the Monte Carlo code should run.

If one is using an input light source which is either a point source or a circle of uniformly distributed light this should be straight forward. However, for any other type of light source an understanding of the Monte Carlo code is needed, and this code is written in c. For example if the input light direction vector is [1 1 1] then the light is going in the positive z direction. It will have no propagation in the x or y directions, contrary to, at least our, intuition which would say that propagation in the positive z direction would be a vector of [0 0 1]. Why it is chosen this way is not known.
The output of the Monte Carlo code is a matrix describing the fraction of light absorbed in each array position given in [W/cm^3/W delivered]. So by multiplying the output matrix with the total input power in Watts you get the power absorbed pr. volume.

2 Code manual

In this section the different files used in the simulation process is listed and described, along with the procedure of running the simulation.

2.1 List of simulation files

The code used for this simulation tool consist of a number of files, which are shown in the table below along with a short description of the file, and the file type. The files are divided into three types: *program*, *function*, and *not used*. Programs are files that have an output, and need to be opened and run to generate the output. This could fx. be generation of the tissue model or simulation of the bioheat equation. Some of these files have a lot of parameters to be set, others only a few. *Functions* files are either Matlab functions or data sets. The former are small scripts that do a specific task when called from another file. These files cannot be run separately, but will automatically be run or loaded from *program* or *parameter* files. If a *function* file contains parameters to be edited, the program should be saved, but not run after the editing. The *not used* files are extra files that could be incorporated in the simulation process, but currently are not. They do not affect the outcome of the simulations, and may be either *functions* or *programs*.

Filename	**Description**	**File type**
maketissue.m	Generates the desired tissue model	Program
3Dmc.c	Monte Carlo simulation in c code	Program
3Dmc.exe	Monte Carlo simulation program	Program
User_settings.m	For specifying parameters for source and spectrum	Program
Gen_Ap_mcxyz_spectra.m	Weighs a number of Monte Carlo simulations at set wavelengths to a specified broad band spectrum	Program
HeatSim_Ver2.m	Calculates the bioheat equation for a given pulse	Program
Plot_dead.m	Displays the results of the simulations	Program
makeTissueList.m	Lists different tissues types and their properties	Function
dirrec.m	Finds files of a given type in a folder and its subfolders	Function
spectralLIB.m	Contains a library of different tissue properties	Function
Tissue_Prop_Matrix.m	Takes tissue types described by integers and returns them containing e.g. the scattering coefficient at all points instead	Function
lookmcxyz.m	Used instead of Gen_Ap_mcxyz_spectra.m for monochromatic sources	Not used
reportHmci.m	Lists the values of the input file myname_H.mci	Not used
TissuesPropertiesPlots.m	Generates plots of the optical tissue parameters used	Not used

2.2 File description

Here a further explanation of each file is given. Specifically, an overview over which inputs and outputs *program* files have, as well as which parameters should be specified in *parameter* and *program* files are shown. The list does not contain *not used* files, since they are not crucial to the simulation procedure. Further, it only describes the *function* files that contain parameters. A description of the other functions can be found as a header in the files themselves.

maketissue.m

Inputs: makeTissueList.m
Outputs: myname_T.bin, myname_H.mci
Parameters: SAVEON, nm, myname, time_min, zsurf, SC, epd_thick, vessel_thick, vessel_depth, (Nbins, binsize, mcflag, launchflag, xs, ys, zs, xfocus, yfocus, zfocus, radius, waist, ux0, uy0)

This file generates the tissue model. It inputs the different tissue type properties from makeTissueList.m. The depth and thickness of the different tissue types are set using the parameters zsurf, SC, epd_thick, vessel_thick and vessel_depth in the section "CREATE TISSUE STRUCTURE T(y,x,z)". The other parameters are found in the section "USER CHOICES". nm sets the range of wavelengths wished to be simulated, and time_min indicate how long the Monte Carlo simulation should run, for each wavelength. SAVEON can be set to 1 or 0, to respectively save or not save the two output files. If SAVEON = 1, myname specifies the title of the output files, and is made to incorporate the wavelength. As such, if the wavelength range is from 500 nm to 600 nm in steps of 100 nm, the file will output four files, two for each wavelength.
The parameters in parenthesis can be edited, but will likely not need to be. Nbins and binsize controls the number and size of pixels/voxels in the tissue model. The other parameters controls the light source, which is currently set to a collimated, uniform beam over the whole skin surface.
Commented out in the program, in the section "DRAW TISSUE MODEL", is a piece of code that generate a figure of the model.

makeTissueList.m

Inputs: spectralLIB.m
Outputs: -
Parameters: mua, mus, g, gg, HC, TC, B, S, W, M, musp500, fray, bmie

This file contains the physical properties for different tissue types. Note that the parameters should be set for each different tissue type. HC is the heat capacity and TC the thermal conductivity. The other parameters relate to the optical properties of the tissue. g or gg for some tissue types, is the anisotropy factor, which is between -1 and 1. The scattering coefficient is either set directly as mus, or calculated using musp500, fray ad bmie. When that is the case, musp500 is the scattering at 500 nm, fray is the fraction of the scattering judged to come from Rayleigh scattering, and bmie is the exponent when calculating the Mie scattering. The formula $musp500 * (fray * (nm/500).^{-}4 + (1 -$

$fray) * (nm/500).^{-}bmie)$ is then used to calculate the scattering as a function of wavelength, first term coming from Rayleigh scattering (when particle size is much smaller than the wavelength of light), and the last term from Mie scattering.
For the absorption coefficient, this can either be set directly as mua, or calculated using B, S, W and M. These are numbers between 0 and 1, and correlates to how much haemoglobin (B), water (W) and melanin (M) there is in the tissue, as well as the fraction of haemoglobin that is oxygenated (S). The scattering coefficients for these three tissue types are given as a function of wavelength in the loaded data set spectralLIB.m

3Dmc.c and 3Dmc.exe

Inputs: myname_T.bin, myname_H.mci
Outputs: myname_F.bin, myname_props.m
Parameters: -

This is the code used to run the Monte Carlo simulation. Note that this is code in the language C, not Matlab like the rest of the files. That means that the the code is edited in the 3Dmc.c file, but run using the file 3Dmc.exe.
The code for the Monte Carlo simulation uses the parameters chosen in maketissue.m, and as such does not have any parameters to change. The procedure of the Monte Carlo simulation is that it inputs a number of photons (depending on the time set in maketissue.m for the simulation) and simulate their propagation through the tissue, taking scattering and absorption into account. The output is a matrix containing the distribution of where the photons have been absorbed. It is important that the time set for the simulation is long enough that the distribution is representative.

User_settings.m

Inputs: filename.txt
Outputs: Input_spectrum.m
Parameters: dnm, nm_min, nm_max, dF, F_min, F_max, pulse, filename

This file input a spectrum defined in a text file, which contains two columns of numbers, first one being wavelengths, second one being intensity. Since the spectrum is normalized in Gen_Ap_mcxyz_spectra.m to fit with the desired fluence, it does not need to be normalized already in the text file. filename sets the name of the spectrum file to be loaded. The wavelength range and step size is found using nm_min, nm_max and dnm. nm_min and nm_max need to be within the range of the spectrum, and dnm needs to be an multiple of the spectrum wavelength step size. Thus, a spectrum defined from 400 nm to 700 nm in steps of 2 nm, cannot have nm_min lower than 400 nm, nm_max higher than 700nm, neither can it have dnm of fx. 5 nm, since it is not a multiple of 2. The fluence of the pulse can be set in a similar way using F_min, F_max and dF. The simulation that Input_spectrum is input in, will be run once for each fluence set here. This gives the possibility of running several simulations with

different input fluences in one go. if F_min = F_max, only one fluence is set. The pulse time of the light source is set in the parameter pulse.

Gen_Ap_mcxyz_spectra.m

Inputs: myname_F.bin, Input_spectrum.m
Outputs: savename.m
Parameters: savename, myname

This file inputs the spectrum settings specified in User_settings.m, as well as the distribution of absorbed photons from the Monte Carlo simulation. For each wavelength of the spectrum, the absorption distribution is loaded from the Monte Carlo simulation, and then weighed to fit with the intensity of the given wavelength in the spectrum. The input spectrum is normalized and then multiplied with the desired fluency to give a absolute number of photons absorbed at each wavelength. The output is a the weighed distribution of absorbed photons as a function of wavelength.
This file can be used for a monochromatic source if the spectrum specified is monochromatic, or the range of wavelengths is set to a single one.

HeatSim_Ver2.m

Inputs: Input_spectrum, savename.m
Outputs: save_name.m
Parameters: savename, save_name, duration_after, dt, Temp_initial, image_interval, Bad_Temp

This file calculates the bioheat equation for each input fluence set in User_settings.m. This means calculating the heat diffusion of the absorbed photons found via the Monte Carlo simulation. That uses the assumption that the light propagation and absorption happens on a much faster scale than the heat diffusion. The simulation uses the following equation:

$$u(z,t) = 10 \sum_{n_z=1}^{\infty} \frac{2}{n_z \pi} \left(\cos\left(\frac{3}{8} n_z \pi\right) - \cos\left(\frac{5}{8} n_z \pi\right) \right) \exp\left(-\frac{n_z^2 \pi^2}{L_z^2} kt \right) \sin\left(\frac{n_z \pi z}{L_z} \right)$$

The program calculates the equation in discrete time steps of dt. For the pulse duration, the heat sources in form of the absorbed photons are assumed to be constant, after which the heat source disappears and the calculations are done for the amount of time set in duration_after. The initial temperature of the skin is set uniformly to Temp_initial. The simulation is stopped if at any point the temperature Bad_Temp is reach, which is high that is unphysical. image_interval gives the time steps at which the hat profiles are saved. save_name sets the name of the output files.

Plot_dead.m

Inputs: save_name.m
Outputs: -
Parameters: save_name, SC_thick, Temp_dead

This file visualizes the results of the bioheat equation. This is done by inputting the heat distribution as a function of time from the HeatSim_Ver2.m. The maximum temperature at each pixel is then found, and if it exceeds Temp_dead, that pixel is shown as dead on the a figure of the tissue. Further the percentage of damage, calculated as number of dead pixel over the total amount of pixel for respectively blood, vessel wall and epidermis is calculated. It is possible to animate the heat distribution, though that part of the code is put after a 'break' in the code, and as such have to be run manually. Note that as the code is now, only the last of the fluences will get animated even then.

2.3 Simulation description

There are several steps to the simulation. For a given tissue model, step 1 and 2 can be made just once, after which the simulation can be started at step 3, for each new spectrum distribution to be simulated.

1. **Creating the tissue model:**
   Use the maketissue.m file to make the tissue model. Set user settings, mainly containing information about the light source, distribution as well as the desired wavelengths and simulation time for the Monte Carlo simulation. This is done in the section "USER CHOICES". Also set the thickness of tissue layer under the section "CREATE TISSUE STRUCTURE". The optical properties of each tissue type can be edited in the file makeTissueList.m.
   Saves the files myname_T.bin and myname_H.bin used in the Monte Carlo simulation. Saves those files for each wavelength. 'myname' is a string specified in maketissue.m and contains the specific wavelength in the end of the string, _xxx.

2. **Run the Monte Carlo simulation:**
   To run the Monte Carlo simulation open the command prompt, navigate to the position of 3Dmc.exe. Call "3Dmc.exe myname" to run the Monte Carlo simulation. This needs to be done separately for each specified wavelength. Uses myname_T.bin and myname_H.bin as input files.
   Generates the files myname_F.bin and myname_props.m.

3. **Define user settings:**
   In the file user_settings.m, set the specified parameters. This inputs a file filename.m which contain the spectrum of the light source. User parameters include wavelength ranges of the input spectrum as well as the pulse time and fluences. A range of fluences can be set.

4. **Define the light source:**
   Set input parameters in and run Gen_Ap_mcxyz_spectra.m to generate the input for HeatSim_Ver2.m. Input parameters are savename and myname, as well as what is defined in user_settings.m.

A file savename.m is generated. Several files will be generated, corresponding to the number of fluences are set in the user_setting.m file

5. **Run the bioheat equation:**
   Run HeatSim_Ver2.m to calculate the heat diffusion during and after illumination. Inputs the files savename.m and save_name. Outputs a number of files save_name.m, corresponding to the number of fluences set in user_settings.m

6. **Visualizing the effect:**
   Run the file Plot_dead.m, to visualize the damage and effect on the skin of the given simulation. Last part of the code is for making an animation of the heat distribution over time. This part is put after a 'break' in the code, and as such has to be run manually using ctrl+enter. Note that the pulse fluence the animation will contain is the last one of the range.